

ESP8266 meshed network exploration results as of 19-03-2019.

The “ESP8266MQTTMesh” library provides the possibility to have a radio mesh created automatically. This network is created at the radio link level, not at the WiFi level. Each node (esp module) is given a unique id. For esp8266 modules this is the unique chip id. This chip id is an integer value. This id is used to create the radio mesh.

The individual nodes are not aware of the structure of the network.

One of the nodes of the mesh establishes a WiFi connection with Mosquitto.

The data exchange mesh is created at the level of Mosquitto. The structure of the network resides on Mosquitto.

This is done by a clever way of concatenating topics and subtopics and relaying messages if required. The id is part of the topic which is used to create the meshed network at MQTT level.

You can't subscribe to an arbitrary topic. The default topic that is subscribed to is “esp8266-in/” though this can be changed during initialisation.

The specified callback will be called whenever a topic of “esp8266-in/” is received (or if “esp8266-in/broadcast” is received) is the node's chipid in hex (not zero padded) I.e. if your chipid was 123456 the expected topic would be “esp8266-in/1e240/”.

By a small adaptation of the library this chip-id can be set in the software of the node to the rocnet node-id. The rocnet node-id is then used instead of the standard chip-id.

After performing some preliminary tests the result is as follows:

the outgoing topic can be set to “rocnet/rocnet node-id/. . .”

the incoming topic can be set to “rocnet/rocnet node-id/. . .”

the incoming topic can also be set to “rocnet/broadcast/....”

Because the information is inserted by means of a pre-compiler statement, it is not possible to change this value after compilation. Each node requires separate compilation of the software.

The data which are transferred via the mesh are of the type char or string. Byte data are not supported. Rocnet can handle “HEXA”. This protocol facilitates the transfer of byte data using ASCII characters. The only drawback is that each byte requires two ASCII characters for transmission. This is not a problem because of the very small messages used with the rocnet protocol.

Some live testing without Rocrail have to be performed to confirm these findings.

Test setup

The mesh contains 2 switch decoders and 2 sensor decoders. Outside the mesh one rocail “emulator” is used. The software for those component is tailor made for testing purposes.

Each switch decoder has an LED. Each sensor decoder has a push button. The emulator will have two pushbuttons and two LEDs.

One test will be to see if it is possible to turn the LEDs on both decoders on and of with the push buttons of the emulator.

The second test will be to see if it is possible to switch the LEDs of the emulator on and of with the pushbuttons of the switch decoders.